

FIG. 2

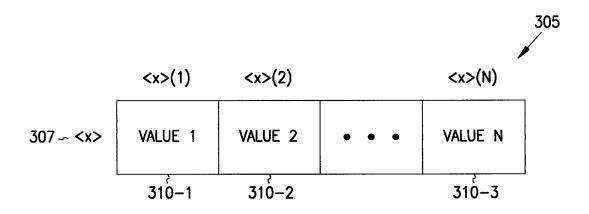


FIG. 3A

				35
355∽ <y>(1)</y>	VALUE 1	LOCATION 1	TIMESTAMP 1	
360 ∽ <y>(2)</y>	VALUE 2	LOCATION 2	TIMESTAMP 2	
l	•	•	•	I
365∽ <y>(N)</y>	VALUE N	LOCATION N	TIMESTAMP N	

FIG. 3B

```
$
          450
          405
                                                                           print ("average %f\n", sum/count);
                                                     sum += p->value;
p = p->tail;
                    p = aList;
sum = 0;
count = 0;
while (p != NULL) {
                                                count += 1;
```

FIG. 4

```
200
                   550
                                                       p = aList;

565~ while (p != NULL) { 560

if (count<while> != 1) {

555 printf(", ");
                                                                                                       printf("%d", p->value);
p = p->tail;
                    505
                          p - aList;
firstTime = true;
while (p != NULL) {
   if (firstTime) {
      firstTime = false;
   } else {
      printf (", ");
   }
                                                                                                                                 printf("%d", p->value);
p = p->tail;
```

FIG. 5

```
900
                                                                                  for (i = 0; i < ARRAY_SIZE; i++) { 665 \sim x = a[i]; 660 } printf ("Max is %f\n", max<x>);
                           ,
650
                                                                    607 - int max = a[0];

for (i = 0; i < ARRAY_SIZE; i++) {
   if (max < a[i]) {
      max = a[i];
                              605
                                                                                                                                                                              printf ("Max is %f\n", max);
```

FIG. 6

```
while (!eof(aFile)) {
765~ x = read(aFile);
}

755~1
for (i = 0; i < length<x>; i++) {
printf("%d %f", i, <x>[i]);
760~2
760~2
                         750
                                                                                                                                    for (i = 0; i < list.length(); i++) { printf("%d %f", i, list[i]);
                             705
                                                          706 ~ intVector list;

while (!eof(aFile)) {

int x = read(aFile);

list.append(x);
```

FIG. 7

```
805
                                806
printf ("x %s been assigned", length \langle x \rangle ==0? "has not": "has");
                                     808
                                                              810
                                               811
    printf ("%d warning message(s) printed", length<warning>);
                                                       813
                                                              815
           printf ("d lines of input read", length<gets>);
                                                               820
                               821
                              reset<x>
                                   823
                                                               825
                             826
min<x-y>
                                   828
```

FIG. 8

```
900
                                                                                 950
                                                                  x = f(x);
                                       while (. . .) {
if (test(x)) {
965~ label:
                 905
                                                                                                printf ("count: %d", counter);
                              906 -- int counter = 0;

while (. . .) {
    if (test(x)) {
        counter += 1;
        x = f(x);
```

FIG. 9

```
1000
                               1050
                                                                                                                                                                                                                       1055-1 \scriptstyle{\sim} count<br/>
posTest.then>, \scriptstyle{\sim} 1060-1
                                                                                                                                                                                               printf ("then: %d, else: %d",
                                                                                                                                                                                                                                        1055-2 ~ count<posTest.else>);
                                                                                                                                                                                                                                                                1060-2
                                                                                                        if (x > 0) {

1065-1 \sim y = dx + dy;

\frac{1065-2}{2} \sim y = dx - dy;
                                                                                               posTest:
                                  1005
                                                                                                                                                                                                                    printf ("then: %d, else: %d",
                                                           1006 -- int thenCount = 0;

1007 -- int elseCount = 0;

if (x > 0) {

thenCount += 1;

y = dx + dy;

} else {

elseCount +=1;
                                                                                                                                                                                    y = dx - dy;
                                                                                                                                                                                                                                             thenCount,
elseCount);
```

FIG. 10

```
1100
    1150
                   1105
           1106 \sim int limit = 0;
              x = f(0);
do {
```

FIG. 11

```
p = aList;
while (p != NULL) {
    x = p.head();
    match:
    found = equal(p.head, key);
    if (found) break;
    p = p.tail();
    }
}
print (searching required %d probes\n", length<match:equal>);
```

FIG. 12

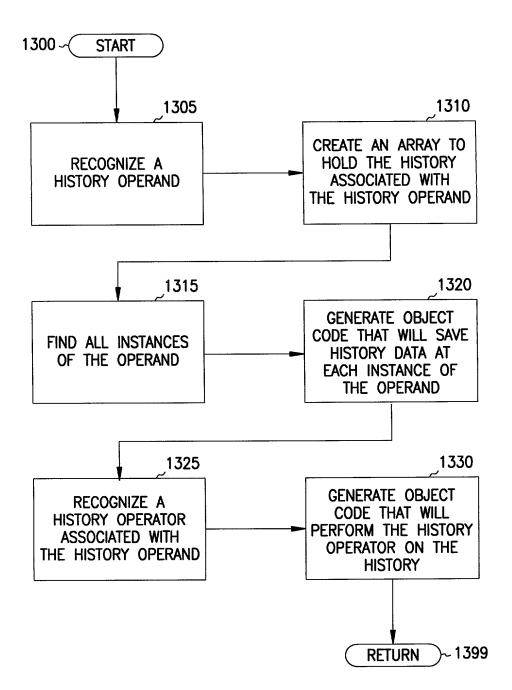


FIG. 13